



Conflict coloring problems: (theory, complexity, algorithm), and application to multi-resolution modeling in structural bioinformatics

Frédéric Havet, Dorian Mazauric, Thi Viet Ha Nguyen, Frédéric Cazals

► To cite this version:

Frédéric Havet, Dorian Mazauric, Thi Viet Ha Nguyen, Frédéric Cazals. Conflict coloring problems: (theory, complexity, algorithm), and application to multi-resolution modeling in structural bioinformatics. 2022. hal-03666380

HAL Id: hal-03666380

<https://inria.hal.science/hal-03666380>

Preprint submitted on 12 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Conflict coloring problems: (theory, complexity, algorithm), and application to multi-resolution modeling in structural bioinformatics

FRÉDÉRIC CAZALS*, Centre Inria d'Université Côte d'Azur, France

FRÉDÉRIC HAVET, CNRS, Université Côte d'Azur, I3S, Inria

DORIAN MAZAURIC, Centre Inria d'Université Côte d'Azur, France

VIET HA NGUYEN, CNRS, Université Côte d'Azur, I3S, Inria

Given a graph $G = (V, E)$, a color set $C(v)$ for each vertex $v \in V$, a bipartite graph between color sets $C(u)$ and $C(v)$ for every edge $uv \in E$, CONFLICT COLORING consists in deciding whether there exists a conflict coloring, that is a coloring in which $c(u)c(v)$ is not an edge of the bipartite graph. CONFLICT COLORING is motivated by high-resolution determination of molecular assemblies. The graph represents the subunits and the interaction between them, the colors are the given conformations, and the edges of the bipartite graphs are the incompatible conformations of two subunits.

In this paper, we first establish the complexity dichotomies (polynomial vs NP-complete) for CONFLICT COLORING and its variants. We provide some experiments in which we build instances of CONFLICT COLORING associated to *Voronoi diagram* in the plane, and we then analyse the existences of a solution related to parameters used in our experimental setup.

CCS Concepts: • **Theory of computation** → **Design and analysis of algorithms; Computational complexity and cryptography; Randomness, geometry and discrete structures**; • **Applied computing** → **Life and medical sciences**;

ACM Reference format:

Frédéric Cazals, Frédéric Havet, Dorian Mazauric, and Viet Ha Nguyen. 2022. Conflict coloring problems: (theory, complexity, algorithm), and application to multi-resolution modeling in structural bioinformatics. 1, 1, Article 1 (January 2022), 22 pages.

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Graph coloring problems. Given a graph $G = (V, E)$ such that each vertex v is equipped with a color set $C(v)$, a *coloring* c of G is an assignment of colors to all vertices in V such that $c(v) \in C(v)$ for all $v \in V$. If for any edge $uv \in E$, we are given a bipartite graph between color sets $C(u), C(v)$ (whose edges imply incompatible pairs of colors), a *conflict coloring* c is a coloring in which $c(u)c(v)$ is not an edge of the bipartite graph. The problem for deciding whether there exists a conflict coloring of a given graph is called CONFLICT COLORING. Observe that the classical problem (PROPER) COLORING is the case when the bipartite graph between color sets of any edge uv form a perfect matching of edges joiniong the same color for u and for v (the sets of colors are the same for every vertex). In fact, a given bipartite graph

*The work of Frédéric Cazals has been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

Manuscript submitted to ACM

for each edge gives a binary relation between the two endpoints of this edge. Hence CONFLICT COLORING belongs to (binary) Constraint Satisfaction Problem (CSP). These two problems are formally presented in Section 2. CSP is a very general formalism to specify many problems in real life such as map-coloring, robotic [19], scheduling [6],... It then has been widely studied mostly in Artificial Intelligence and Operation Research. The NP-completeness of CSP motivates the interest in finding techniques to improve solvers (e.g. CPLEX [20], and Choco [13]) which try to solve CSP instances as fast and efficient as possible. It is advantageous to use available studies on CSP and apply to CONFLICT COLORING in particular and other combinatorial problems in general in application-side works.

Motivation from computational structural biology: structural determination of macromolecular assemblies. All biological phenomena (cognition, immune response, metabolism, etc) are based on interactions between biomolecules, namely proteins or nucleic acids [8, 22]. In the sequel, we focus on *molecular assemblies* which involve of the order of tens of subunits, each typically involving hundreds of amino acids (for proteins) or bases (for nucleic acids). The first step in understanding the biological functions operated by such assemblies is to elucidate their structure, ideally at the atomic level—in which case the individual atomic positions are known precisely. However, because each atom has three Cartesian coordinates, such systems involve hundreds of thousands of degrees of freedom, posing experimental and modeling difficulties. On the experimental side, despite recent progresses in structure determination using X-ray crystallography and cryo-electron microscopy, large and/or flexible assemblies still cannot be solved routinely. On the modeling standpoint, the very large dimensionality precludes *ab initio* approaches, so that a multi-resolution - two stage approach based on low-resolution and then high-resolution models is called for. In a first step, a low-resolution model coding pairwise contacts between subunits is built [2, 3]. Such a model is coded by an *interaction graph* whose vertices correspond to the individual subunits. In a second step, this graph is leveraged. To see how, assume that a finite set of high-resolution *conformations* has been computed for each individual subunit, and that a *scoring function* assessing the quality of the contact between two interacting conformations is available. In applying the scoring function to all pairs of interacting conformations (as coded in the interaction graph), the goal is to identify the individual conformations of the subunits.

The approach just described is the Discrete Optimization of Multiple INteracting Objects (DOMINO) approach introduced in [16]. It can be formalized as a graph coloring problem by considering a conformation of a subunit as a color of the corresponding vertex. Hence, the (decision) coloring problem consists in finding a coloring such that the sum of the weights is at least a given lower bound. The DOMINO problem has mostly been studied from biology point of view and some works are related to computational approaches. Despite the NP-completeness of DOMINO [17], there are methods to solve this problem with heuristic approaches, using tree decomposition [16, 24] or linear programming [25]. In this article, we study a new approach of DOMINO in which the score function is relaxed, CONFLICT COLORING.

Our contributions and paper overview. In this article, we study CONFLICT COLORING, the new approach of DOMINO in which the weight function is simplified (binary relations). This problem and variants are formally introduced in Section 2. CONFLICT COLORING is a particular case of Constraint Satisfaction Problem (CSP) which have studied in many communities. We provide some related works of its computational complexity approaches. In Section 3, we establish a dichotomy of complexity (P / NP-complete) for CONFLICT COLORING and its variants. Finally, in Section 4, we provide selected experiments on geometric instances of CONFLICT COLORING, based on *Voronoi diagrams/Delaunay triangulations* in the plane. This setup provides a simple and parameterizable scenario for the low/high-resolution

reconstruction of molecular assemblies, making it possible to study the dependency between parameters of the model and the hardness/existence of solutions.

2 MODELS AND RELATED PROBLEMS

Let $G = (V, E)$ be an interaction graph representing a given assembly in which V represents the set of subunits and E is the set of contacts between them. We label the set of conformations for each subunit to a color set of the corresponding vertex in V . We assume that every vertex has the same number of colors which is an integer k , hence the set of colors is $[k] = \{1, 2, \dots, k\}$. For every $uv \in E$, the scoring function provides the edge-weighted complete bipartite graph K_{uv}^k which is defined as follows: $V(K_{uv}^k) = \{(u, i), i \in [k]\} \cup \{(v, j), j \in [k]\}$, $E(K_{uv}^k) = \{(u, i)(v, j), i, j \in [k]\}$, and the weight function $w_{uv}^k : E(K_{uv}^k) \rightarrow \mathbb{R} \cup \{-\infty\}$. We denote by $w^k = \{w_{uv}^k, uv \in E\}$ the set of weight functions over all the edges $uv \in E$.

Hence, for every vertex $v \in V$, the vertex set $\{(v, i), i \in [k]\}$ represents the k possible conformations of the subunit associated with v . For every $uv \in E$, the weight $w_{uv}^k((u, i)(v, j))$ represents the *score* between the i -conformation of u and the j -one of v . Meaning by the scoring function, the higher the score, the better the connection between the two associated subunits.

In general, the number of colors is not the same for all vertices. However, we can easily change it so that all vertices have the same number k of colors: it suffices to add extra “fake” colors and to assign weight $-\infty$ to all edges incident with these colors in the graph K_{uv}^k . It explains our assumption of the same number of colors for all vertices.

2.1 Conflict coloring

An important constraint, from computational structural biology, consists in having a lower bound $\lambda \in \mathbb{R}$ on the weight of every edge $uv \in E$. More precisely, the coloring c must satisfy $w_{uv}^k((u, c(u)), (v, c(v))) \geq \lambda$ for every $uv \in E$. This constraint may imply a stable structure of the macromolecular assembly, that is without “bad connection” between two subunits.

To take into account such a constraint, we define a *conflict k -graph* M_{uv}^λ for every edge $uv \in E$ as follows. M_{uv}^λ is the subgraph of K_{uv}^k such that $(u, c(u))(v, c(v)) \in E(M_{uv}^\lambda)$ if and only if $w_{uv}^k((u, c(u)), (v, c(v))) < \lambda$. We denote by $\mathcal{M}^\lambda = \{M_e^\lambda \text{ for all } e \in E\}$, the *set of conflict k -graphs* (k -SoC). An edge $uv \in E$ is \mathcal{M}^λ -*conflicting* for a k -coloring c if $(u, c(u))(v, c(v)) \in E(M_{uv}^\lambda)$, otherwise it is \mathcal{M}^λ -*fulfilled*.

The problem is then to solve the following problem with $\mathcal{M} = \mathcal{M}^\lambda$.

FULLFILL COLORING.

Input: A graph $G = (V, E)$, an integer k , a k -SoC \mathcal{M} , and a positive integer q .

Question: Does there exist a k -coloring c of G with at least q \mathcal{M} -fulfilled edges ?

An important particular case of this problem is when $q = |E|$. In that case, the problem, called **CONFLICT COLORING**, is to determine whether G is \mathcal{M} -*conflict colorable*, that is whether it admits a k -coloring with no \mathcal{M} -conflict edges. Such a coloring is called an \mathcal{M} -*conflict coloring*. This notion of conflict coloring was introduced by Dvořák and Postle [7] and Fraigniaud et al. [9].

We shall consider the restrictions of FULLFILL COLORING, and CONFLICT COLORING to the case when k is fixed. These are called FULLFILL k -COLORING, and CONFLICT k -COLORING, respectively.

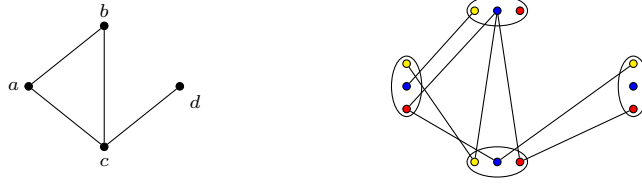


Fig. 1. **Example of conflict coloring.** A graph G on four vertices (left) and a 3-SoC \mathcal{M} of G (right) with 3 colors yellow, blue and red; $M_e \in \mathcal{B}(3, 2)$: for every $e \in E(G)$ contains two edges. An \mathcal{M} -Conflict Coloring c of G is $c(a) = c(c) = c(d) = \text{"blue"}$, $c(b) = \text{"red"}$. Coloring c is a solution of $\mathcal{B}(3, 2)$ -CONFLICT 3-COLORING.

We finally define variants of FULFILL COLORING and its restrictions, in which \mathcal{M} is contained in some restricted family \mathcal{B} of bipartite graphs. For example, \mathcal{B} -CONFLICT k -COLORING is the restriction of CONFLICT k -COLORING in which each conflict k -graph M_e must belong to \mathcal{B} . We denote by $\mathcal{B}(n, m)$ the class of subgraphs of $K_{n,n}$ with m edges. Figure 1 describes an instance of $\mathcal{B}(3, 2)$ -CONFLICT 3-COLORING and a solution. When \mathcal{B} is the family of matchings (i.e. bipartite graphs with maximum degree 1), \mathcal{B} -CONFLICT k -COLORING is known as *correspondence coloring* or *DP-coloring*. Since it is a generalization of the classical notions of coloring and list coloring, it has been extensively studied over the past five years (see e.g., [4, 7, 15, 18]).

2.2 Relations to CSP

CSP is a very general problem an instance of is defined by a triple $(\mathcal{V}, \mathcal{D}, C)$ where $\mathcal{V} = \{V_1, \dots, V_n\}$ is a set of variables, $\mathcal{D} = \{D_1, \dots, D_n\}$ is a domain of values and $C = \{C_1, \dots, C_m\}$ is a set of constraints. Each variable V_i can take a value in its domain D_i . Every constraint C_j is a triple (r_j, t_j, R_j) where r_j is a positive integer called *arity* of the constraint, t_j is an r_j -tuple of variables and R_j is an r_j -ary relation on t_j . An *evaluation* satisfies a constraint (r_j, t_j, R_j) if the variables of t_j are assigned values which satisfies R_j . An evaluation which satisfies all constraints is called a *solution*. Observe that H -COLORING and CONFLICT COLORING are particular cases of binary CSP where all constraints are binary relations (on variable domains corresponding to the vertex colors).

A finite constraint language Γ is a finite domain and a finite set of relations over this domain. $\text{CSP}(\Gamma)$ is the constraint satisfaction problem where instances are only allowed to use constraints in Γ .

A bipartite conflict graph M in a k -SoC \mathcal{M} can be converted to a binary relation R_M with $(c, c') \in R_M \iff (c, c') \notin E(M)$ for all $c, c' \in [k]$. As a consequence we can reduce an instance of k -CONFLICT COLORING to the instance of CSP with variables $V(G)$ on domain $[k]$ and the set of constraints $\{R_{M_{uv}}, uv \in E(G)\}$ corresponding to the conflict edges. However, when \mathcal{M} is restricted to a family \mathcal{B} of bipartite graphs, the constraint language $\Gamma_{\mathcal{B}} = \{R_M, M \in \mathcal{B}\}$ gives a problem $\text{CSP}(\Gamma_{\mathcal{B}})$ with more instances than \mathcal{B} -CONFLICT k -COLORING, in other words \mathcal{B} -CONFLICT k -COLORING is a restriction of $\text{CSP}(\Gamma_{\mathcal{B}})$. Indeed, in $\text{CSP}(\Gamma_{\mathcal{B}})$ we can choose arbitrary constraints among the relations and, in particular, multiple constraints on the same couple of variables; let say they are represented as conflict graphs M_1, \dots, M_ℓ . So this would correspond in \mathcal{B} -CONFLICT k -COLORING to a conflict graph for the same pair of vertices which is the *union* of M_1, \dots, M_ℓ (because CSP is a *conjunction* of constraints). From this reasoning we deduce the following:

LEMMA 2.1. *If \mathcal{B} is closed by union, i.e. $M, M' \in \mathcal{B} \implies (M \cup M') \in \mathcal{B}$, then \mathcal{B} -CONFLICT k -COLORING is equivalent to $\text{CSP}(\Gamma_{\mathcal{B}})$ with $\Gamma_{\mathcal{B}} = \{R_M, M \in \mathcal{B}\}$.*

As a consequence, complexity upper bounds on $\text{CSP}(\Gamma_{\mathcal{B}})$ apply to \mathcal{B} -CONFLICT k -COLORING, and complexity lower bounds on \mathcal{B} -CONFLICT k -COLORING apply to $\text{CSP}(\Gamma_{\mathcal{B}})$. When \mathcal{B} is not closed by union, we cannot immediately apply the complexity lower bound results from $\text{CSP}(\Gamma_{\mathcal{B}})$.

2.3 Our contributions

We first study the computational complexity of FULFILL- k -COLORING, CONFLICT- k -COLORING and their restrictions on graph families $\mathcal{B}(k, m)$ in Section 3.

If for every $uv \in E(G)$, the edge set of M_{uv} is the matching $\{(u, i), (v, i) \mid i \in [k]\}$, then the graph is \mathcal{M} -conflict colorable if and only if it is k -colorable. Therefore k -COLORING is a particular case of CONFLICT k -COLORING. It is well-known that k -COLORING is NP-complete if and only if $k \geq 3$. Henceforth, the problems CONFLICT k -COLORING and FULFILL k -COLORING are NP-complete for all $k \geq 3$. Thus, the problems CONFLICT COLORING and FULFILL COLORING are NP-complete. We prove in Theorem 3.1 that CONFLICT 2-COLORING is polynomial-time solvable. On the other hand, FULFILL 2-COLORING is more general than MAX CUT (decision version). This problem is given a graph and an integer, and aims to determine whether there is a partition of vertices into two disjoint sets such that there number of edges between them is at least the number. It is known to be NP-complete [14]. Therefore FULFILL 2-COLORING is also NP-complete.

We next study $\mathcal{B}(k, m)$ -FULFILL k -COLORING and $\mathcal{B}(k, m)$ -CONFLICT k -COLORING. We prove polynomial vs. NP dichotomy results with respect to the considered values k and m . All those complexity results are summarized in Table 1.

Remark 2.2. *Given a coloring, we can easily check whether it satisfies the problems FULFILL-COLORING or CONFLICT-COLORING in polynomial time. Thus, these two problems are in NP and we only prove the NP-hardness of a NP-complete result.*

	FULFILL k -COLORING	CONFLICT k -COLORING
$k = 1$	polynomial	polynomial
$k = 2$	NP-complete	polynomial
$k \geq 3$	NP-complete	NP-complete
	$\mathcal{B}(k, m)$ -FULFILL k -COLORING	$\mathcal{B}(k, m)$ -CONFLICT k -COLORING
$m = 0$	polynomial	polynomial
$1 \leq m \leq k^2 - 3$	NP-complete	NP-complete
$m \in \{k^2 - 1, k^2 - 2\}$	NP-complete	polynomial
$m = k^2$	polynomial	polynomial

Table 1. Dichotomy results on conflict coloring.

The next section describes experimentation works. We first introduce our setup to build (geometric) instances of CONFLICT COLORING and then present some statistics which we obtain from our experiments. Namely, instances of CONFLICT COLORING are associated with *Voronoi diagrams* in the plane in which we retain as vertices those data points which have a finite Voronoi region. Then, we build a k -SoC \mathcal{M} by generating a set of k (randomly) perturbed polygons of the finite Voronoi polygon of a point, and given an intersection area threshold τ_c , a conflict between two perturbed polygons of adjacent Voronoi regions is when their intersection area is greater than τ_c (detail in Section 4.1).

Hence, a random instance and a threshold τ_c yield a set of conflicts. The obtained statistics provide information on the relationship between the existence of a solution and thresholds which are derived from some local conditions.

We have also studied algorithms for solving the coloring problems and DOMINO. This is not in this paper but can be read in [21].

3 COMPLEXITY OF THE PROBLEMS

The aim of this section is to prove results in Table 1.

3.1 Unrestricted conflict coloring

We first establish the complexity of CONFLICT k -COLORING and FULFILL k -COLORING, and then consider the restrictions of the problems to the family $\mathcal{B}(k, m)$.

CONFLICT 1-COLORING and FULFILL 1-COLORING are trivial: there is only one possible color for each vertex, we just need to check which edges e are fulfilled, that is such that M_e is edgeless.

We now consider CONFLICT 2-COLORING and obtain the following.

THEOREM 3.1. *CONFLICT 2-COLORING is polynomial-time solvable.*

PROOF. Let $G = (V, E)$ be a graph and let M_{uv} be the conflict 2-graph for each edge $uv \in E$. If there is an edge $e \in E$ such that $M_e = K_{2,2}$, then the answer is ‘No’ because e cannot be fulfilled. If there is an edge $e \in E$ such that $M_e = K_{2,0}$ (edgeless graph), then e is always fulfilled. Without loss of generality, we assume that the conflict 2-graph of every edge is neither $K_{2,2}$ nor $K_{2,0}$.

Now, from G and the set of conflict 2-graphs M_{uv} for each edge $uv \in E$, we construct a 2-SAT formula Φ as follows.

(1) For a vertex $v \in V$ and a color in $\{1, 2\}$, create a variable x_v for $(v, 1)$ and \bar{x}_v for $(v, 2)$. So we denote by ℓ_v^i a literal of variable x_v , in which $\ell_v^1 = x_v$ and $\ell_v^2 = \bar{x}_v$.

(2) For an edge $uv \in E$ with M_{uv} , we create a clause gadget $C_{uv} = \bigwedge_{(u,i)(v,j) \in E(M_{uv})} (\bar{\ell}_u^i \vee \ell_v^j)$.

We will prove that G admits a conflict coloring c if and only if there is an assignment ϕ satisfying Φ .

Observe that, for any $uv \in E(G)$, the clause gadget C_{uv} contains a clause for each edge $(u, i)(v, j) \in M_{uv}$ which is the negation of $(\ell_u^i \wedge \ell_v^j)$. It implies that c does not conflict on uv by edge $(u, i)(v, j) \in M_{uv}$ if and only if $\neg(\ell_u^i \wedge \ell_v^j)$ is satisfiable by ϕ . Thus, uv is fulfilled by c if and only if C_{uv} is satisfiable by ϕ .

Therefore, one can determine c from ϕ by taking $c(v) = 1$ (resp. $c(v) = 2$) if $\phi(x_v) = \text{true}$ (resp. $\phi(x_v) = \text{false}$). \square

In contrast to CONFLICT 2-COLORING, FULFILL 2-COLORING generalizes the well-known NP-complete problem MAX CUT, and thus it is also NP-complete. For $k \geq 3$, CONFLICT k -COLORING generalizes the NP-complete problem k -COLORING and so is NP-complete. Hence, FULFILL k -COLORING is also NP-complete.

PROPOSITION 3.2. *CONFLICT k -COLORING is NP-complete for all $k \geq 3$. FULFILL k -COLORING is NP-complete for all $k \geq 2$.*

3.2 Restrictions to the family $\mathcal{B}(k, m)$

Now, we look at the restrictions of those problems when every conflict k -graph M_e is in $\mathcal{B}(k, m)$, that is a bipartite graph with k vertices in each part and m edges. We assume that $k \geq 2$. If $m = 0$ (resp. $m = k^2$), then the conflict k -graphs

are edgeless (resp. complete bipartite) and so every k -coloring fulfills all edges (resp. no edges). Hence $\mathcal{B}(k, 0)$ -FULFILL k -COLORING and $\mathcal{B}(k, k^2)$ -FULFILL k -COLORING are trivially polynomial-time solvable.

When $k = 2$, $\mathcal{B}(2, m)$ -CONFLICT 2-COLORING is in P by Theorem 3.1. We prove the following for $\mathcal{B}(2, m)$ -FULFILL 2-COLORING.

THEOREM 3.3. $\mathcal{B}(2, m)$ -FULFILL 2-COLORING is NP-complete for $m \in \{1, 2, 3\}$.

PROOF. An easy observation is that $\mathcal{B}(2, 2)$ -FULFILL 2-COLORING is NP-complete since it generalizes MAX CUT. Thus, we now prove for graph classes $\mathcal{B}(2, 1)$ and $\mathcal{B}(2, 3)$.

1. $\mathcal{B}(2, 1)$ -FULFILL 2-COLORING is NP-hard.

We give a reduction from MAX-2-SAT problem which is known to be NP-hard [10].

Given a 2-SAT formula Φ with N variables x_1, \dots, x_N and l clauses, each containing exactly two literals, we construct a graph G and the conflict 2-graphs M_e for all $e \in E(G)$ as follows (see Figure 2 for an example):

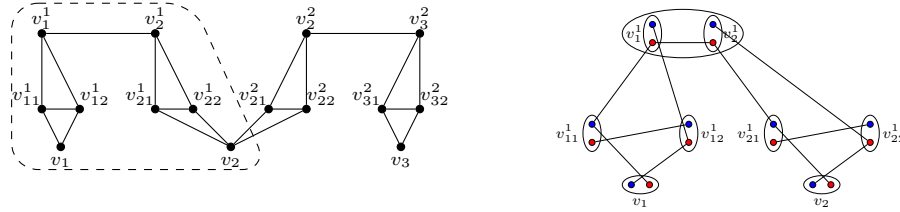


Fig. 2. **An example of the construction for $\Phi = (x_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3)$.** The graph G (left) and the union of the conflict 2-graphs of edges in the dashed box (colors 1, 2 are respectively blue, red).

- For each variable x_t , we create a vertex v_t , and if x_t appears in α_t clauses, then we create three copies $v_{t1}^r, v_{t2}^r, v_t^r$ for each $r \in [\alpha_t]$. We add edges $v_{t1}^r v_t^r, v_{t2}^r v_t^r \in E(G)$; and $(v_{t1}^r, 1)(v_t^r, 2) \in M_{v_{t1}^r v_t^r}$ and $(v_{t2}^r, 2)(v_t^r, 1) \in M_{v_{t2}^r v_t^r}$. For $r \in [\alpha_t]$, we create a cycle on vertices v_t, v_{t1}^r, v_{t2}^r , and for (ordered edge) $uv \in \{v_t v_{t1}^r, v_{t1}^r v_{t2}^r, v_{t2}^r v_t\}$, M_{uv} contains only the edge $(u, 2)(v, 1)$. For convenience, we call the subgraph on the four vertices $v_t, v_{t1}^r, v_{t2}^r, v_t^r$ a v_t^r -graph for every $r \in [\alpha_t]$.
- For each clause, assume that $C = (\ell_1 \vee \ell_2)$ where ℓ_1, ℓ_2 are respectively the r_1 -th occurrence of x_p, r_2 -th occurrence of x_q in C . We add to $E(G)$ the clause edge $e = v_p^{r_1} v_q^{r_2}$, and create its conflict 2-graph M_e as follows:
 - If $\ell_1 = x_p$ and $\ell_2 = x_q$, then $(v_p^{r_1}, 2)(v_q^{r_2}, 2) \in M_e$.
 - If $\ell_1 = x_p$ and $\ell_2 = \bar{x}_q$, then $(v_p^{r_1}, 2)(v_q^{r_2}, 1) \in M_e$.
 - If $\ell_1 = \bar{x}_p$ and $\ell_2 = x_q$, then $(v_p^{r_1}, 1)(v_q^{r_2}, 2) \in M_e$.
 - If $\ell_1 = \bar{x}_p$ and $\ell_2 = \bar{x}_q$, then $(v_p^{r_1}, 1)(v_q^{r_2}, 1) \in M_e$.

We shall prove that Φ admits an assignment which satisfies at least k clauses if and only if there is a 2-coloring c which fulfills at least $10l + k$ edges of G .

Assume that we have an assignment ϕ satisfying at least k clauses of Φ . Then, we take a coloring as follows:

- if $x_t = \text{true}$ (resp. false), then $c(v_t) = 1$ (resp. $c(v_t) = 2$).
- all its copies $v_{t1}^r, v_{t2}^r, v_t^r$ for $r \in [\alpha_t]$ have color $c(v_t)$.

We now check that there are at least $10l + k$ fulfilled edges in G . Indeed, for each satisfied clause, the corresponding clause edge in $E(G)$ is fulfilled, then we have at least k such edges. For each variable x_t , and $r \in [\alpha_t]$, the five edges of

the v_t^r -graph are fulfilled. There are $2l$ such graphs which in total have $10l$ fulfilled edges. Therefore, c fulfills at least $k + 10l$ edges of G .

Conversely, assume there is a coloring which fulfills at least $10l + k$ edges of G . Let c be a coloring that fulfills the maximum number of edges.

The following claim can easily be checked.

Claim 3.3.1. *The five edges in a v_t^r -graph are fulfilled if and only if its four vertices have the same color.*

Claim 3.3.2. *We may assume that c fulfills all the edges of all v_t^r -graphs (for $t \in [N], r \in [\alpha_t]$).*

Proof of claim: Assume a coloring in which there is a v_t^r -graph which has a conflict edge. By Claim 3.3.1, when recoloring all vertices of this graph with a same color, all its edge are fulfilled. Moreover, the only edge that may possibly become conflict with such a recoloring is the clause edge at v_t^r . Hence the new coloring has at least as many fulfilled edges as c , and we can consider it instead of c . \triangleleft

We define ϕ by taking $\phi(x_t) = \text{true}$ (resp. false) if $c(v_t) = 1$ (resp. $c(v_t) = 2$). Since c fulfills all edges of v_t^r -graphs ($10l$ in total), then c fulfills at least k clause edges in G . Thus, ϕ satisfies at least k clauses of Φ .

2. $\mathcal{B}(2, 3)$ -FULFILL 2-COLORING is NP-hard.

We give a reduction from 3-SAT. Given a formula Φ of this problem with N variables x_t , $t \in [N]$ and l clauses C_i , $i \in [l]$, we construct a graph G and M_e for any $e \in E(G)$ as follows (see Figure 3 for an example).

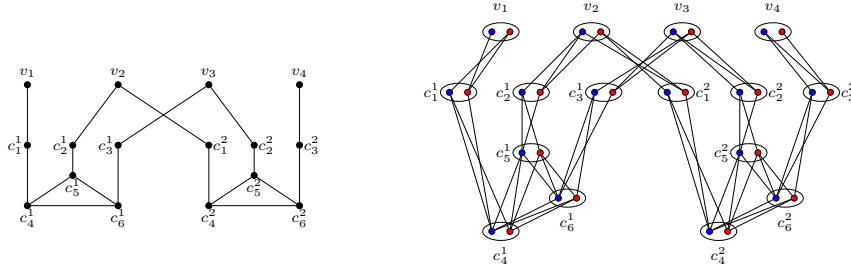


Fig. 3. **An example of the construction in the reduction.** The graph G (left) which is associated to $\Phi = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$ and an example of graphs M_e for $e \in E(G)$. The color 1 (resp. 2) is blue (resp. red).

- For each variable x_t , create a vertex $v_t \in V(G)$.
- For each clause C_i , we create a *clause gadget* X_i which contains six vertices c_1^i, \dots, c_6^i . We add (ordered) edges $c_4^i c_5^i, c_5^i c_6^i, c_6^i c_4^i$ and for each edge uv of them, M_{uv} has three edges which are not $(u, 1)(v, 2)$. If $C_i = (\ell_1 \vee \ell_2 \vee \ell_3)$ where each ℓ_j is a literal of variable x_{t_j} , we then add a set Y_i of *variable-clause edges* $v_{t_1} c_1^i, v_{t_2} c_2^i, v_{t_3} c_3^i$. For $j \in [3]$, $M_{v_{t_j} c_j^i}$ has three edges which are not $(v_{t_j}, 1)(c_j^i, 1)$ (resp. $(v_{t_j}, 2)(c_j^i, 1)$) if $\ell_j = x_{t_j}$ (resp. $\ell_j = \bar{x}_{t_j}$). For convenience, for a clause C_i , the edge set which contains all edges of X_i and Y_i is called *clause-edge set* and is denoted by S_i .

One can easily check the following.

Claim 3.3.3. *For each clause C_i , a coloring fulfills no edges in Y_i if and only if it can fulfill at most three edges in S_i . Furthermore, there exists a coloring which fulfills four edges in S_i in which at least one of them is in Y_i .*

We shall show that there is a truth assignment ϕ satisfying Φ if and only if there is a 2-coloring of G with $4l$ fulfilled edges. The idea is that a variable $\phi(x_t) = \text{true}$ (resp. false) when $c(v_t) = 1$ (resp. $c(v_t) = 2$), and a literal $\ell_j = x_t$ (resp. $\ell_j = \bar{x}_t$) is true in a clause C_i when the edge $v_t c_j^i \in Y_i$ is fulfilled and then v_t is colored 1 (resp. 2).

Assume that there is an assignment ϕ satisfying Φ . We define a coloring c of G by taking $c(v_t) = 1$ if $\phi(x_t) = \text{true}$ or $c(v_t) = 2$ if $\phi(x_t) = \text{false}$, and we complete c such that if C_i is true, then c fulfills four edges of S_i which includes the variable-clause edge of every true literal in C_i ; otherwise, c fulfills three edges in S_i and none of them is in Y_i . This can be done by Claim 3.3.3. Since there are l clauses of Φ which are true, then c fulfills $4l$ edges.

Conversely, assume that we have a 2-coloring c of G with $4l$ fulfilled edges. By Claim 3.3.3, c must fulfill at least one edge in Y_i for any clause C_i . As a consequence, we obtain ϕ by taking $\phi(x_t) = \text{true}$ (resp. false) if $c(v_t) = 1$ (resp. 2), hence ϕ satisfies Φ . \square

Remark 3.4. *MAX CUT can be solved in polynomial time for planar graphs [11]. In contrast, $\mathcal{B}(2, 3)$ -FULFILL 2-COLORING is NP-complete on planar graphs by reducing from PLANAR 3-SAT which is NP-complete [?] with the same reduction in the proof Theorem 3.3-2.*

When $k \geq 3$, we use reductions from k -COLORING to establish the following.

THEOREM 3.5. *Let $k \geq 3$ be an integer. If $1 \leq m \leq k^2 - 3$ then $\mathcal{B}(k, m)$ -CONFLICT k -COLORING is NP-complete, and if $m = \{k^2 - 1, k^2 - 2\}$ then it is polynomial-time solvable.*

The proof is separated into several cases depending on m .

Polynomial cases when $m \in \{k^2 - 1, k^2 - 2\}$. Given a graph G , we shall find a conflict k -coloring c of G if it exists. If $m = k^2 - 1$, in order to fulfill an edge $uv \in E$, we have to take $c(u) = i, c(v) = j$ for the only edge $(u, i)(v, j) \notin M_{uv}$.

If the colors forced at a vertex by ever edge is the same, then we obtain a conflict k -coloring of G . Otherwise, G is not conflict colorable.

If $m = k^2 - 2$, the problem can be solved as follows. Initially, let $\tilde{c}(v) = \emptyset$ for every $v \in V$. We sequentially apply the following rules for all vertices.

Rule 1: For every edge $uv \in E$, if there is $i \in [k]$ such that the two edges $(u, i)(v, j_1), (u, i)(v, j_2) \notin M_{uv}$, then $\tilde{c}(u) := \tilde{c}(u) \cup \{i\}$.

If there is a vertex v such that $|\tilde{c}(v)| \geq 2$, then there is no coloring c because v needs two colors to fulfill all its incident edges. Otherwise, let $S = \{v \in V \mid |\tilde{c}(v)| = 1\} \subseteq V$.

We sequentially apply the following for all vertices $V \setminus S$.

Rule 2: For every vertex v with all its neighbors $\{u_1, \dots, u_q\}$ and for every color $i \in [k]$ such that $(v, i)(u_j, c_j) \notin M_{vu_j}$ for all $j \in [q]$ and some $c_j \in [k]$, then we set $\tilde{c}(v) := \tilde{c}(v) \cup \{i\}$.

Observe that $|\tilde{c}(v)| \leq 2$ for every $v \in V$. If there is a vertex $v \in V$ such that $\tilde{c}(v) = \emptyset$, then there is no conflict coloring for G because there is no color for v that allows to fulfill all its incident edges. Without loss of generality, we assume that $G[V \setminus S]$ is connected (otherwise, we repeat the following for all the connected components).

There are two cases:

- (i) If there is a vertex $v \in V \setminus S$ such that $|\tilde{c}(v)| = 1$, let $(v_1 = v, v_2, \dots, v_t)$ be the ordering obtained by a breadth-first search algorithm in $G[V \setminus S]$. By construction of S , and $V \setminus S$, this ordering gives a unique color for all vertices of $V \setminus S$, that is $c(v_1) = \tilde{c}(v_1)$, and $c(v_i) = c_i$ with c_i the unique color such that $(v_{i-1}, c(v_{i-1}))(v_i, c_i) \notin M_{v_{i-1}v_i}$ for every $i \in \{2, \dots, t\}$. Finally, once the colors $c(v)$ have been determined for all vertices $v \in V$, we have to verify if all the edges of E are fulfilled.

- (ii) If $|\tilde{c}(v)| = 2$ for every $v \in V \setminus S$, take any vertex $v \in V \setminus S$ and fix one color $c(v) \in \tilde{c}(v)$ and apply the procedure above in (i). If there is a coloring that fulfills all edges, then it terminates. Otherwise, apply the procedure in (i) for the other color in $\tilde{c}(v)$. Hence either we find a conflict k -coloring or G has no such a coloring.

The NP-completeness when $m \in [k^2 - 3]$. We denote by $[a, b]$ for $a \leq b$ the set of integers $\{a, a+1, \dots, b\}$. For an edge uv , we reuse the notation K_{uv}^k which is the complete bipartite with the vertex set $V(K_{uv}^k) = \{(u, i), i \in [k]\} \cup \{(v, j), j \in [k]\}$ and $E(K_{uv}^k) = \{(u, i)(v, j), i, j \in [k]\}$.

We reduce from k -COLORING and the proof includes the three following cases.

Case 1: $\frac{k}{2} \leq m \leq (k-1)^2$:

Given a graph $G = (V, E)$ of k -COLORING, we construct, for $\mathcal{B}(k, m)$ -CONFLICT k -COLORING, a graph $G' = (V', E')$ and a k -SoC \mathcal{M} as follows (Figure 4):

- For each $v \in V$, we add to V' a vertex v' , and for each edge $uv \in E$, add k vertices $a_1^{uv}, \dots, a_k^{uv}$. We set $E'_{uv} = \{ua_i^{uv}, va_i^{uv} \mid i \in [k]\}$ and add it to E' .
- For any $uv \in E$ and $i \in [k]$, $M_{ua_i^{uv}}$ contains $t = \min\{m, k-1\}$ edges $(u, i)(a_i^{uv}, j)$ for $j \in [t]$ and $t' = m - (k-1)$ (if $t' > 0$) edges of B_u where B_u is the complete bipartite graph whose vertex sets are $\bigcup_{j \in [k] \setminus \{i\}} \{(u, j)\}$ and $\bigcup_{j \in [2, k-1]} \{(a_i^{uv}, j)\}$. The conflict graph $M_{va_i^{uv}}$ contains t edges $(v, i)(a_i^{uv}, k), \dots, (v, i)(a_i^{uv}, k-t+1)$ and t' (if $t' > 0$) edges of B_v where B_v is the complete bipartite graph whose vertex sets are $\bigcup_{j \in [k] \setminus \{i\}} \{(v, j)\}$ and $\bigcup_{j \in [2, k-1]} \{(a_i^{uv}, j)\}$.

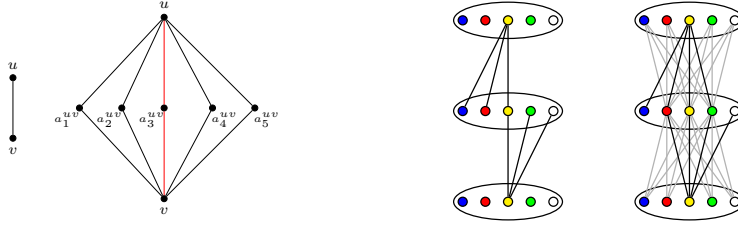


Fig. 4. An example of the construction when $k = 5$. An edge $uv \in E$ and its set E'_{uv} (left). The conflict k -graphs of red edges ua_3^{uv}, va_3^{uv} when $m = 3$ (third graph); and if $m > 4$, there are $t = 4$ black edges and the $t' = m - k + 1$ other edges are chosen among grey edges (rightmost).

Observe that by the construction of the set \mathcal{M} , the value m is at most $(k-1)^2$, because for any edge $uv \in E$ and $i \in [k]$, $M_{ua_i^{uv}}$ and $M_{va_i^{uv}}$ contains at most $k-1 + |E(B_u)| = k-1 + |E(B_v)| = (k-1)^2$ edges.

We shall prove that G is k -colorable if and only if G' is \mathcal{M} -conflict colorable.

Assume that c is a k -coloring of G , then we can construct a coloring c' of G' as follows: $c'(v') = c(v)$ for any $v \in V$, $c'(a_i^e) = k$ for all $e \in E, i \in [k]$. We can check easily that c' fulfills all edges of G' .

Conversely, there is an \mathcal{M} -conflict coloring c' of G' . Observe that if $c'(u) = i$ (for some $i \in [k]$), then $c'(a_i^{uv}) \in [t, k]$. It implies that $c'(v) \neq i$ because otherwise $t \geq k/2$, so va_i^{uv} is not fulfilled. Thus, $c'(u) \neq c'(v)$ and we then obtain a coloring c of G by taking $c(v) = c'(v')$ for every $v \in V$.

Case 2: $1 \leq m < \frac{k}{2}$:

We need the following lemma.

Manuscript submitted to ACM

LEMMA 3.6. *There is a graph family $\{G(k, m) \mid k, m \geq 1\}$ such that $G(k, m)$ is not $\mathcal{B}(k, m)$ -conflict k -colorable.*

PROOF. Given $m \geq 1$, we shall construct a family $\{G(k, m) \mid k \geq 1\}$ and a k -SoC \mathcal{M} by induction on k . Let $G(k, m) = (V(k, m), E(k, m))$. Consider the following cases.

- (i) For $m = k^2$, let $G(k, m)$ on 2 vertices a_1, a_2 and it has one edge $a_1 a_2$. And let the conflict k -graph $M_{a_1 a_2} = K_{a_1 a_2}^k$ (Figure 5).

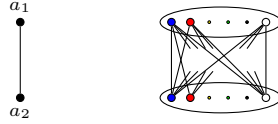


Fig. 5. Graph $G(k, k^2)$.

- (ii) For $k^2 > m \geq k$, $G(k, m)$ has $k + 1$ vertices a, a_1, \dots, a_k and k edges aa_i for all $i \in [k]$. For each of them, the conflict k -graph M_{aa_i} contains k edges $(a, i)(a_i, j)$ for all $j \in [k]$, and $m - k$ other edges are chosen among the other edges of $K_{aa_i}^k$ (Figure 6).

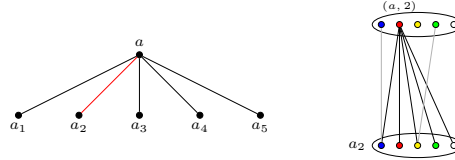


Fig. 6. **An example of graph $G(5, m)$ with 5 colors.** The conflict 5-graph of red edge aa_2 , when $m = 5$, M_{aa_2} contains five black edges $(a, 2)(a_2, i)$ for all $i \in [5]$; and when $m = 7$, then M_{aa_2} contains these black edges and two grey edges which are in the remaining edges of $K_{aa_2}^5$.

- (iii) For $k > m$, $G(k, m)$ is built as follows (Figure 7). We add $k + 1$ vertices a, a_1, \dots, a_k to $V(k, m)$, and add k edges aa_i for all $i \in [k]$ to $E(k, m)$. Now, for each vertex a_i , add a set X_i of $|G(k - m, m)| - 1$ new vertices, and let a copy C_i of the graph $G(k - m, m)$ on $a_i \cup X_i$. Then, connect a to all vertices of this copy C_i .

Now, we construct a set of conflict k -graphs for $G(k, m)$. For each edge ab (for every $b \in C_i, 1 \leq i \leq k$), the conflict k -graph M_{ab} contains m edges $(a, i)(b, j)$ for all $j \in [m]$. In each copy C_i , we apply $(k - m)$ -SoC \mathcal{M}' of graph $G(k - m, m)$ for $k - m$ colors $\{m + 1, \dots, k\}$, if $m \geq (k - m)^2$, then the conflict $(k - m)$ -graph M_{uv} for $uv \in C_i$ contains the complete bipartite on two sets $\{(u, m + 1), \dots, (u, k)\}, \{(v, m + 1), \dots, (v, k)\}$ and $m - (k - m)^2$ remaining edges are taken in K_{uv}^m .

Trivially, graphs in case (i) have no \mathcal{M} -conflict coloring.

In case (ii), $G(k, m)$ has no conflict coloring, since choosing any color $i \in [k]$ for a implies that edge aa_i is not fulfilled.

In case (iii), by the construction, colors of a_1, \dots, a_k are not in $[m]$. However, if we choose a color for each a_i in $[m + 1, k]$, then it can not fulfill all edges of the copy C_i because $G(k - m, m)$ is not conflict colorable with its $(k - m)$ -SoC. Thus, the graphs of this case have no conflict coloring. \square

Now, we give a reduction from k -COLORING problem to $\mathcal{B}(k, m)$ -CONFLICT k -COLORING for $1 \leq m < k/2$.

Given a graph $G = (V, E)$, we construct a graph $G' = (V', E')$ and a k -SoC \mathcal{M} as follows.

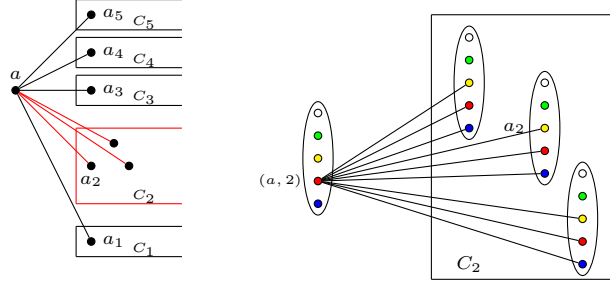


Fig. 7. **An example of graph $G(5, 3)$.** A copy $C_i = G(2, 3)$ for all $i \in [5]$. The conflict k -graph of red edges and copy C_2 (right): vertex $(a, 2)$ connects three first vertices $(b, 1)$, $(b, 2)$, $(b, 3)$ (presented as blue, red, yellow) and the set 2-SoC of $G(2, 3)$ is applied to vertices $(b, 4)$, $(b, 5)$ (presented as green, white) for any $b \in V(C_2)$.

- Let A be a copy of V and add A to V' . For each edge $uv \in E$ with $u', v' \in A$ copies of u, v , we add k copies $C_1^{uv}, \dots, C_k^{uv}$ of the graph $G(k - 2m, m)$, and add edges from each u', v' to all vertices in these k copies. Let $V'_{uv} = \{u', v', V(C_1^{uv}), \dots, V(C_k^{uv})\}$.
- For any induced subgraph $G'[V'_{uv}]$, $u' \in A$, $a \in V(C_i^{uv})$ and $i \in [k]$, the conflict k -graph $M_{u'a}$ has m edges $(u, i)(a, 1), \dots, (u, i)(a, m)$; and the conflict k -graph $M_{v'b}$ for any $b \in V(C_j^{uv})$, $j \in [k]$ has m edges $(v, j)(b, k), \dots, (v, j)(b, k - m + 1)$. In each C_i^{uv} , we apply the $(k - 2m)$ -SoC of the graph $G(k - 2m, m)$ for $k - 2m$ colors $[m + 1, k - m]$.

If G has a k -coloring c then we have c' for G' as follows: for each V'_{uv} , let $c'(v') = c(v)$, $c'(u') = c'(a) = c(u)$ for all $a \in \bigcup_{i \in [k] \setminus c(u)} V(C_i^{uv})$ and $c'(a) \in [k, k - m + 1]$ for all $a \in V(C_{c(u)}^{uv})$. Hence, c' is a conflict coloring of G' .

Conversely, assume G' has a k -coloring c' , then we prove that $c'(u') \neq c'(v')$ for all $u', v' \in A \cap V'_{uv}$. Observe that for any $a \in V(C_{c'(u')}^{uv})$, its color is not in $[k, k - m + 1]$ because of the construction of $C_{c'(u')}^{uv}$. Moreover, if $a \in C_i^{uv}$ has color $j \notin \{m + 1, \dots, k - m\}$, then (a, j) is either adjacent to (u', i) in the conflict k -graph $M_{u'a}$ or (v, i) in $M_{v'a}$ for $i \in [k]$. Thus $c'(u') \neq c'(v')$. Therefore, we take $c(v) = c'(v')$ for all v' the copy of v and it is a k -coloring of G .

Case 3: $(k - 1)^2 < m \leq k^2 - 3$.

We use an induction on $k^2 - m$ as follows. The base case is when $k = 3$ and $1 \leq m \leq 6$. From these two cases above, it suffices to prove for $m \in \{5, 6\}$.

Claim 3.6.1. $\mathcal{B}(3, m)$ -CONFLICT k -COLORING is NP-complete for $m = \{3, 4, 5, 6\}$.

The inductive result is obtained by the following.

Claim 3.6.2. If $\mathcal{B}(k, m)$ -CONFLICT k -COLORING is NP-complete, then so is $\mathcal{B}(k', m')$ -CONFLICT k' -COLORING for $k' > k$ and $k'^2 - m' = k^2 - m$.

Claim 3.6.1 and Claim 3.6.2 imply the result for Case 3. Indeed if $\mathcal{B}(k, m)$ -CONFLICT k -COLORING is NP-complete for $1 \leq m \leq (k - 1)^2$, then so is $\mathcal{B}((k + 1), m')$ -CONFLICT k -COLORING by Claim 3.6.2 for m' such that $m' = 2k + 1 + m$, thus $2k + 2 \leq m' \leq (k + 1)^2 - 3$.

To complete Case 3, it remains to prove Claim 3.6.1 and Claim 3.6.2.

Proof of Claim 3.6.1. We give a reduction from 3-COLORING. Given a graph $G = (V, E)$ of this problem, we construct for $\mathcal{B}(3, m)$ -CONFLICT k -COLORING, a graph $G' = (V', E')$ and a set of conflict 3-graphs \mathcal{M} as follows (see Figure 8).

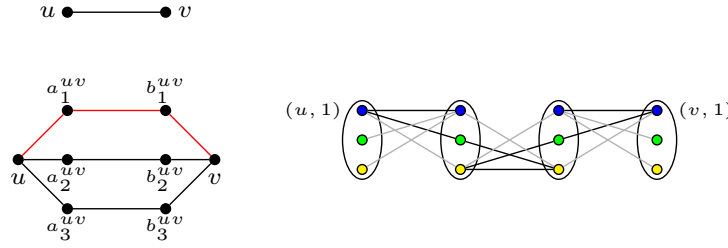


Fig. 8. **Construction of the reduction.** For each edge $uv \in E(G)$, there is a gadget \mathcal{G}^{uv} . And an example on red edges is the structure of their conflict 3-graphs: it is obligatory that black edges are in conflict 3-graphs while gray ones are not. We obtain that $c(u) = 1$ if and only if $c(v) \neq 1$.

- For each edge $e = uv \in E$, add a gadget \mathcal{G}^e of 8 vertices $u', v', a_1^e, b_1^e, a_2^e, b_2^e, a_3^e, b_3^e$ to V' where u', v' are respectively copies of u, v ; and we add to E' edges $ua_i^e, a_i^e b_i^e, b_i^e v$ for all $i \in [3]$.
- For each induced subgraph $G'[\mathcal{G}^{uv}]$, we construct the conflict 3-graphs of edges $u'a_i^{uv}, a_i^{uv} b_i^{uv}, b_i^{uv} v'$ for any $i \in [3]$ as follows:
 - $M_{u'a_i^{uv}}$ contains two edges $(u', i)(a_i^{uv}, 1), (u', i)(a_i^{uv}, 2)$ and any $m - 2$ other edges is in $K_{u'a_i^{uv}}^3$ except three edges $(u', i)(a_i^{uv}, 3), (u', j)(a_i^{uv}, 1)$ for $j \neq i$.
 - $M_{a_i^{uv} b_i^{uv}}$ contains three edges $(a_i^{uv}, 2)(b_i^{uv}, 3), (a_i^{uv}, 3)(b_i^{uv}, 2), (a_i^{uv}, 3)(b_i^{uv}, 3)$ and any $m - 3$ other edges in $K_{a_i^{uv} b_i^{uv}}^3$ except two edges $(a_i^{uv}, 1)(b_i^{uv}, 3), (a_i^{uv}, 3)(b_i^{uv}, 1)$.
 - $M_{b_i^{uv} v'}$ contains two edges $(b_i^{uv}, 1)(v', i), (b_i^{uv}, 2)(v', i)$ and any $m - 2$ other ones in $K_{b_i^{uv} v'}^3$ except three edges $(b_i^{uv}, 3)(v', i), (b_i^{uv}, 1)(v', j)$ for $j \neq i$.

By the construction of \mathcal{M} , each conflict graph $M_e \in \mathcal{M}$ needs at least three conflict edges and at least three edges not in M_e . Thus, $3 \leq m \leq 6$.

We prove now G is 3-colorable if and only if there is c' which fulfills all edges of G' .

If G has a k -coloring c , then the coloring c' of G' defined by taking $c'(u') = c(u), c'(v') = c(v), c'(a_i^{uv}) = 3, c'(b_i^{uv}) = 1$ for all \mathcal{G}^{uv} fulfills all edges of G' .

Conversely, assume there is a k -coloring c' which fulfills all edges of G' . Observe that $c'(u) \neq c'(v)$ for any $u, v \in \mathcal{G}^{uv}$ because by the construction of $G'[\mathcal{G}^{uv}]$, if $c'(u') = c'(v') = i$, then three edges $u'a_i^{uv}, a_i^{uv} b_i^{uv}, b_i^{uv} v'$ are not all fulfilled. Therefore, we define a coloring c of G by taking $c(v) = c'(v')$ for any $v' \in E'$ copy of $v \in E$. \triangleleft

Proof of Claim 3.6.2. Let a graph G and a k -SoC \mathcal{M} for $\mathcal{B}(k, m)$ -CONFLICT k -COLORING. We will construct for $\mathcal{B}(k', m')$ -CONFLICT k' -COLORING, a graph $G' = (V', E')$ and \mathcal{M}' , as follows:

- Let G' be a copy of G (i.e. V', E' are respectively copies of V and E).
- For each edge $u'v'$ of G' , $M'_{u'v'}$ contains the following: a copy of the conflict graph M_{uv} for colors in $[k]$, the complete bipartite graph between two sets $\{(u, i) \mid i \in [k + 1, k']\}$ and $\{(v, j) \mid j \in [k']\}$, and the complete bipartite graph between two sets $\{(u, j) \mid j \in [k']\}$ and $\{(v, i) \mid i \in [k + 1, k]\}$.

If c is a coloring which fulfills all edges of G , then setting $c'(v') = c(v)$ for all $v' \in V'$ copy of $c \in V$, we obtain a coloring c' which fulfills all edges of G' .

Conversely, assume there is a coloring c' which fulfills all edges of G' . Consider that for any $u'v' \in E'$, (v, i) is adjacent to (u, j) for all $i \in [k + 1, k']$ and $j \in [k']$, then $c'(v') \in [k]$ for all $v' \in V'$. Thus the coloring c of G defined by $c(v) = c'(v')$, for $v \in V$, fulfills all edges of G . \blacktriangleleft

Cases 1, 2, and 3 imply that $\mathcal{B}(k, m)$ -FULFILL k -COLORING is NP-complete for $k \geq 3$ and $1 \leq m \leq k^2 - 3$. Moreover, by using the same construction in Claim 3.6.2, we make a reduction from $\mathcal{B}(2, 4 - i)$ -FULFILL k -COLORING to $\mathcal{B}(k, k^2 - i)$ -FULFILL k -COLORING for $i \in \{1, 2\}$. Thus $\mathcal{B}(k, k^2 - i)$ -FULFILL k -COLORING is NP-complete when $i \in \{1, 2\}$.

This completes the proof of Theorem 3.5.

4 EXPERIMENTS

4.1 Geometric random instances of conflict coloring

Geometric model. As a test set, we consider instances associated with Voronoi diagrams in the plane. This geometric setting is especially interesting for several reasons: (i) the number of nodes (cells in the Voronoi diagram) can be tuned, and so is the number of colors (ii) the combinatorial structure of the Voronoi diagram can be used to infer local thresholds/constraints.

Given a set of 2D data points, recall that the Voronoi diagram (VD) is the partition of the plane into convex regions, such that all points in one region have a data point as nearest neighbor [5]. The 2D VD has a dual structure known as the Delaunay triangulation (DT), which (generically) consists of vertices, edges and triangles.

Consider the DT of a dataset P of N points drawn uniformly at random in a square of side $\sqrt{N} * 100$. (This ensures that the average density of points is the same for any value of N , see Remark 4.2 below). Using this geometric setting, we define geometric instances of the conflict coloring problem as follows (Figure 9, Figure 10):

- (Nodes) The nodes of the graph correspond to the data set P of N points. More precisely, we retain as vertices those data points have a *finite Voronoi region* (there are N' such vertices), or equivalently, which do not lie on the convex hull of the data points.
- (Conformations/colors) Consider a data point $p \in P$ and its Voronoi region $V_r(p)$ – the index r stands for *reference*. Let v be a Voronoi vertex of $V_r(p)$. (Note that all such points are finite since p is not on the convex hull.) We take a random perturbation of v in $\mathcal{D}(v, \delta) \setminus V_r(p)$ where $\mathcal{D}(v, \delta)$ is the disc centered at v and radius δ . The set of all such perturbed Voronoi vertices defines a random (perturbed) polygon for p . The set of all perturbed regions (polygons) for point p , denoted $C[p]$, defines its colors.
- (Conflict) Consider two Voronoi regions $V(p) \in C[p]$ and $V(q) \in C[q]$ of two neighboring points in DT. We compute the surface area of the intersection of these two regions, denoted $ia(V(p), V(q))$ for *intersection area*. The regions $V(p)$ and $V(q)$ are termed in conflict if $ia(V(p), V(q)) \geq \tau_c$, for some threshold τ_c .

Summarizing, a random instance is defined by the four tuple: $I = (P, \delta, C, \tau_c)$ with $P = \{p_1, \dots, p_N\}$ a set of N points, δ the radius used to perturb the Voronoi vertices, c the number of colors/randomly perturbed Voronoi regions, and τ_c the threshold to define conflicts.

Remark 4.1. Recall that a polygon is termed simple if two of its edges do not intersect, except possibly at their endpoints. In this work, we use polygon representation provided by the Computational Geometry Algorithm Library [1], see https://doc.cgal.org/Manual/3.1/doc_html/cgal_manual/Polygon/Chapter_main.html. For the sake of simplicity, in sampling conformations, we retain simple polygons only.

Relationship to the structural biology problem. The geometric model just introduced mimics the DOMINO strategy mentioned in Introduction for the reconstruction of macro-molecular assemblies: the low resolution information representing individual contacts between subunits corresponds to edges of the Delaunay triangulation; the conformations of subunits are provided by the perturbed Voronoi regions; the function scoring the interactions between conformations is replaced by the surface area of the intersection between perturbed Voronoi regions.

Practically, our algorithms will be tested on real systems involving of the order of ten subunits—see typical examples in [2, 3]. We just note here that preparing a given assembly for such an experiment is a significant endeavor in itself, requiring three main steps: (i) completion of the structural models of the individual subunits (experimental structures often involve gaps); (ii) conformational exploration of the individual subunits – a very challenging task to obtain sufficiently diverse conformational ensembles, and (iii) choice of a suitable scoring function, which may combine ab-initio energy terms from classical force fields, but also data driven terms obtained from various experimental techniques (e.g. cross-linking to enforce known vicinities between amino acids from different subunits). Such tasks are beyond the scope of this paper.

In the sequel, we therefore use the Delaunay-Voronoi geometric model to study the dependency between its parameters and the hardness/existence of solutions.

4.2 Parameters monitored

A random instance together with a threshold τ_c yield a set of conflicts. We wish to get insights on the relationship between the value of τ_c derived from *local sufficient conditions*, and the existence of global solutions.

Local constraints based on min intersection areas. We use the topological information contained in the DT to derive sufficient conditions on the existence of *local* solutions. More specifically, we consider three nested local levels:

- EDGES. For an edge $p_i p_j$ of the DT, the reference regions $V_r(p_i)$ and $V_r(p_j)$ share a Voronoi edge, and the perturbed regions may overlap. For this edge, we compute the minimum intersection area $\tau_2(p_i p_j)$ of their perturbed regions, which ensures the existence of a local solution, this can be formalized as follows:

$$\tau_2(p_i p_j) = \min_{\substack{V_i \in \mathcal{C}[p_i] \\ V_j \in \mathcal{C}[p_j]}} ia(V_i, V_j). \quad (1)$$

The max value obtained over all edges is denoted $\tau_2^{Max-Min}$.

- TRIANGLES. Consider a triangle $(p_i p_j p_k)$ (face) in DT. In a manner analogous to edges, we compute the minimum value $\tau_3(p_i p_j p_k)$ required have a local solution for this triple. Formally:

$$\tau_3(p_i p_j p_k) = \min_{\substack{V_i \in \mathcal{C}[p_i] \\ V_j \in \mathcal{C}[p_j] \\ V_k \in \mathcal{C}[p_k]}} \max\{ia(V_i, V_j), ia(V_i, V_k), ia(V_j, V_k)\}. \quad (2)$$

The max value obtained over all triangles is denoted $\tau_3^{Max-Min}$.

- STARS. Finally, we consider the star centered at each data point p_i , that contains p_i and its set of neighbors $N(p_i)$ in the DT with finite dual Voronoi cell. We compute the minimum value $\tau_*(p_i)$ of the threshold, which ensures the existence of a local solution of the star. Formally:

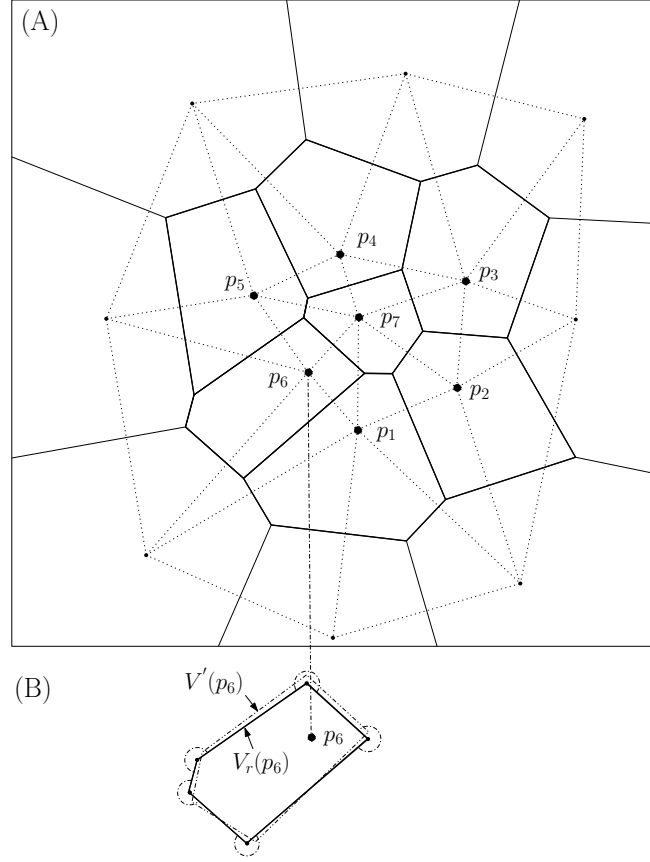


Fig. 9. **Random instance of color matching defined from perturbations of a 2D Voronoi diagram.** (A) In a random 2D Voronoi diagram, points with a finite Voronoi cell (bold cells) are retained – p_1, \dots, p_7 in the example; each such point corresponds to the vertices in the considered graph. The dotted line segments represent the Delaunay triangulation. (B) To create the *colors* associated with a node p of the graph, for each Voronoi vertex v of the Voronoi cell $V_r(p)$, a perturbed point of p is uniformly chosen at random in $\mathcal{D}(v, \delta) \setminus V_r(p)$.

$$\tau_*(p_i) = \min_{V_i \in \mathcal{C}[p_i]} \max_{p_j \in N(p_i)} \min_{V_j \in \mathcal{C}[p_j]} ia(V_i, V_j). \quad (3)$$

The max value obtained over all stars is denoted $\tau_*^{Max-Min}$.

The values $\tau_2(p_i p_j)$, $\tau_3(p_i p_j p_k)$ and $\tau_*(p_i)$ provide local necessary conditions. Observe that, in order to have a global solution for an instance, the threshold δ_c must be at least all these local values. We therefore define

$$\tau_{loc}^{Max-min} = \max\{\tau_2^{Max-Min}, \tau_3^{Max-Min}, \tau_*^{Max-Min}\}. \quad (4)$$

Local constraints based on average intersection areas. In the previous equations, one uses the most stringent constraint for an edge / triangle / star. Consider the case where one increases the number of colors: the min values observed for an edge / triangle / star is expected to decrease. Which means in turn that in increasing the number of colors, fewer solutions are expected overall. Therefore, we also consider a variant of the problem where the local thresholds are

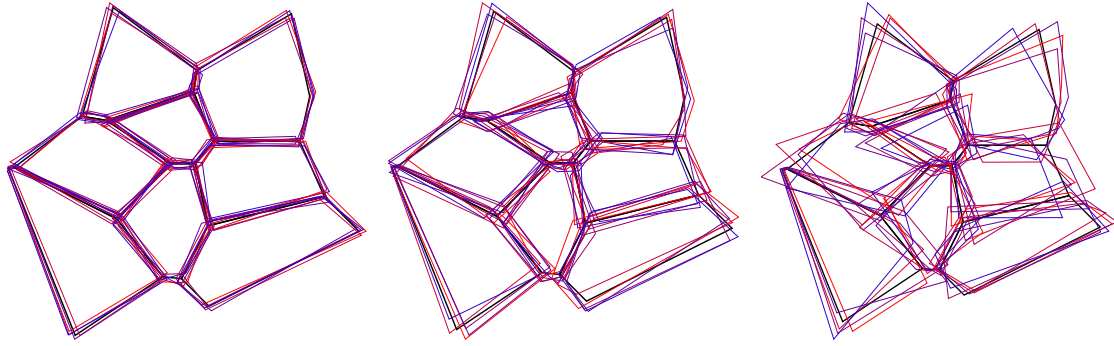


Fig. 10. **Instances corresponding to the construction of Figure 9 upon varying the perturbation radius δ .** Perturbed random polygons of the reference Voronoi regions (thick black polygons) given a data points of size $N = 15$ after generating with perturbed radius $\delta = 0.5$ (left), $\delta = 1$ (middle) and $\delta = 2$ (right) respectively and $c = 5$.

defined from mean intersection areas. Replace the (bold) min by mean in Equations (1), (2), (3) yield the following:

$$\tau_{loc}^{Max-Mean} = \max\{\tau_2^{Max-Mean}, \tau_3^{Max-Mean}, \tau_*^{Max-Mean}\}. \quad (5)$$

Let m be some multiplicative factor – that will differ for τ_c^{\min} and τ_c^{mean} . Using the previous quantities, we define global values of τ_c as follows:

$$\begin{cases} \tau_c^{\min} = m \times \tau_{loc}^{Max-Min}, \\ \tau_c^{\text{mean}} = m \times \tau_{loc}^{Max-Mean}. \end{cases} \quad (6)$$

Remark 4.2. In sampling data points in a square of side $\sqrt{N} * 100$, the expected area for each VD region is 10^4 , the threshold values are in unit and we would mention also in brackets the fraction of a threshold value over the expected area, for example $\tau_{loc} = 100$ (1% of the expected area of a VD region).

4.3 Results

Setup. We consider two complementary analysis: the fraction of solutions in the entire search space, and the existence of at least one solution.

The following values are used in our experiments. Random instances are generated with parameters $N = 15$, $\delta = 1$.

For each instance, we compute the values of τ_c^{\min} and τ_c^{mean} . We retain random instances whose $\tau_{loc}^{Max-Min} \leq 100$ (thus at most 1% of the expected area of a VD polygon). The number of colors used in the experiments varies in the set $C \in \{5, 10, 15, 50, 100\}$. The multiplicative factors vary $m \in m_1 = \{1.01, 1.5, 2, 3, \dots, 10\}$ for τ_c^{\min} and $m \in m_2 = \{0.1, 0.4, 0.7, \dots, 2.2, 3, 5\}$ for τ_c^{mean} .

4.3.1 Conflict coloring is hard.

Statistic of interest. A combinatorial problem may display a phase transition [12], with instances moving from easy to hard when the parameters evolve. We study the hardness of conflict coloring in this respect, using the fraction of valid configurations $f_{c,m}$, namely fraction of valid configurations over the entire search space $\prod_{p_i \in P} C[p_i]$. We compute the

value of $f_{c,m}$ (given the values of τ_c^{\min} and τ_c^{mean}), by enumerating all these configurations. We repeat the calculation $R = 100$, and make a violin plot of the values obtained¹.

Results. The results show a marked difference for values of τ_c^{\min} vs. τ_c^{mean} (Figure 11). For the former, values of $f_{1,01}$ and f_2 are very small, and overall, the distribution of $f_{c,m}$ is shifted towards smaller values when the number of colors c increases. This is sound, as increasing the number of colors is expected to decrease the intersection area, whence the value of $\tau_{\text{loc.}}^{\text{Max-Min}}$. On the other hand, $\tau_{\text{loc.}}^{\text{Max-Mean}}$ being defined from average values, one expect its value to be less sensitive to the number of colors (Figure 15). This lesser sensitivity is obvious from the violin plots associated with τ_c^{mean} , since these are much less sensitive to the number of colors.

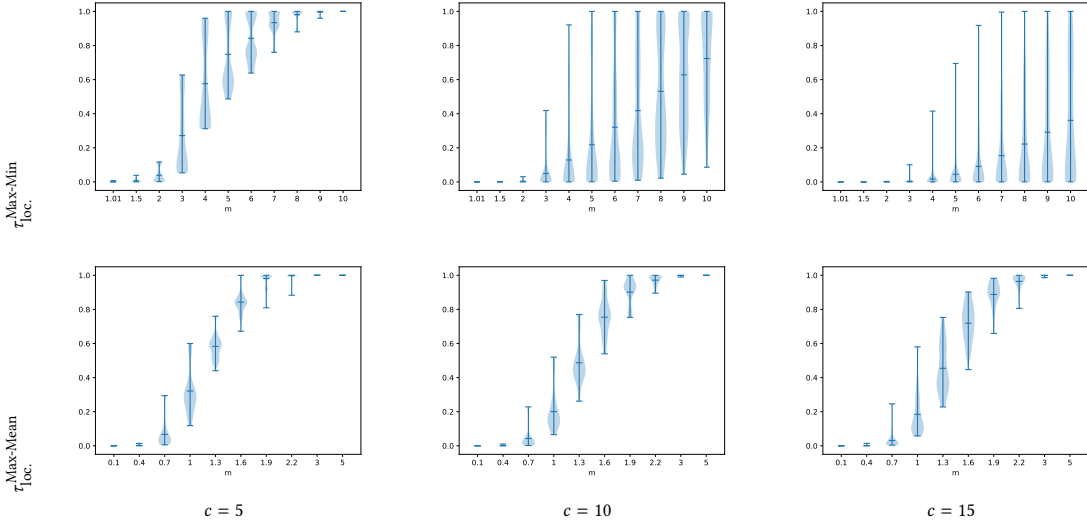


Fig. 11. Violin plots for the fraction $f_{c,m}$ of valid configurations in $\prod_{p_i \in P} C[p_i]$, for fixed values of the number of colors $c \in \{5, 10, 15\}$; each sub-plot features violins where each of which is corresponding to a multiplicative value in m_1 for $\tau_{\text{loc.}}^{\text{Max-Min}}$, or m_2 for $\tau_{\text{loc.}}^{\text{Max-Mean}}$. Each violin plot corresponds to $R = 100$ random instances. These experiments involve $N' = 8$ finite Voronoi regions.

4.3.2 Existence of one solution.

Statistic of interest. Our second analysis focuses on the existence of at least one solution, depending of our global thresholds τ_c^{\min} and τ_c^{mean} (Equation 6). We run our algorithm on $R = 100$ instances, and define the success rate as $s_{c,m} = \text{\#cases with one solution} / R$.

Results. Statistics on $s_{c,m}$ complement those on $f_{c,m}$. Consider first the multiplicative factors m . It clearly appears that $s_{c,m}$ increases when increasing m , for both $\tau_{\text{loc.}}^{\text{Max-Mean}}$ and $\tau_{\text{loc.}}^{\text{Max-Min}}$ (Figure 12). Consider next the number of colors c . While the statistic $s_{c,m}$ hardly depends on the number of colors c for $\tau_{\text{loc.}}^{\text{Max-Mean}}$, it does so for $\tau_{\text{loc.}}^{\text{Max-Min}}$. Namely, in using $\tau_{\text{loc.}}^{\text{Max-Min}}$, $s_{c,m}$ decreases when increasing the number of colors. More precisely, our experiment shows that for

¹ To run the algorithm, we use Choco-a CSP solver see [13].

$c \in \{5, 10, 15, 50, 100\}$, the average of $\tau_{loc.}^{Max-mean}$ is respectively around (120, 100, 97, 90, 91) and (38, 40, 11, 1.3, 0.05) for $\tau_{loc.}^{Max-min}$. It explains the statistics of $s_{c,m}$ (Figure 12) and also $f_{c,m}$ (Figure 11).

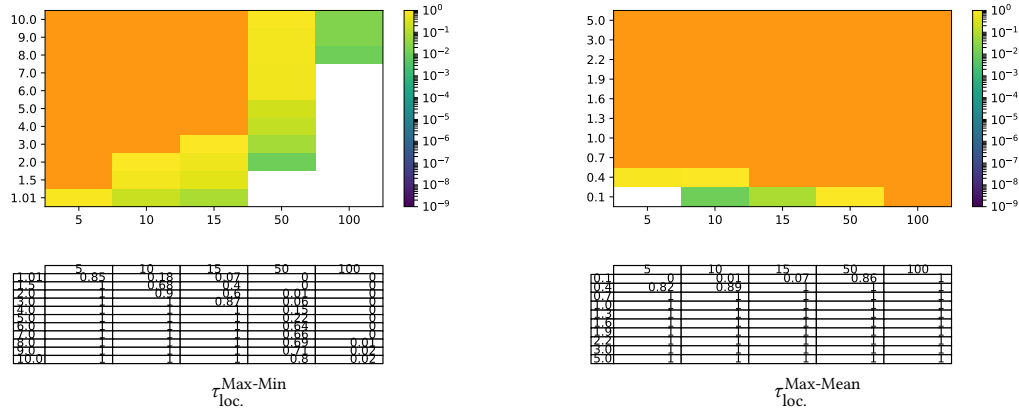


Fig. 12. Heat map representing the fraction $s_{c,m}$ of instances with at least one solution, for fixed values of the number of colors c (horizontal axis) and the multiplicative factor m (vertical axis). In the heat-map and the Tables: rows (resp. columns) correspond to values of c (resp. m). A total of $R = 100$ random instances were used. White color infers value 0.

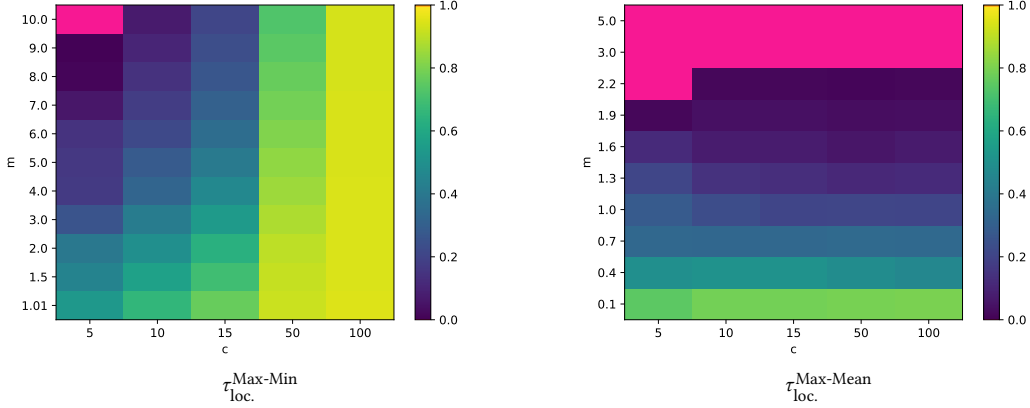


Fig. 13. Heat map representing the fraction of conflict edges, averaged over all edges of the Delaunay triangulation. (Note that for a Delaunay edge, there are at most c^2 conflict edges.) Total run of $R = 100$. Pink color maps to values in $[0, 10^{-3}]$.

4.4 Running times

We provide the running times of checking the existence of one solution for $\tau_{loc.}^{Max-min}$ and $\tau_{loc.}^{Max-mean}$, see Figure 14. Clearly, the running augments when increasing c in general. Figure 14 also show that the time lines corresponding to values m in the left plot are quite similar. In the other hand, those in the right plot tend to 0 when m increases which is

to say that instances get easier. Hence, on our range of m which is approximately behaved as a linear function, the difficulty of an instance does depend for $\tau_{loc}^{Max-mean}$ but does not for $\tau_{loc}^{Max-min}$.

On computational complexity point of view, CONFLICT COLORING is very hard which the running time for solving the problem is exponential to the size of the input. However, the statistics show that the running time does not explode for c big enough and N moderate. Indeed, they are less than 0.25 seconds for $N = 15$ (with $N' = 8$) and $c \leq 100$ (Figure 14). Hence, it is possible to solve the problem CONFLICT COLORING in practice (with instances of tens of subunits).

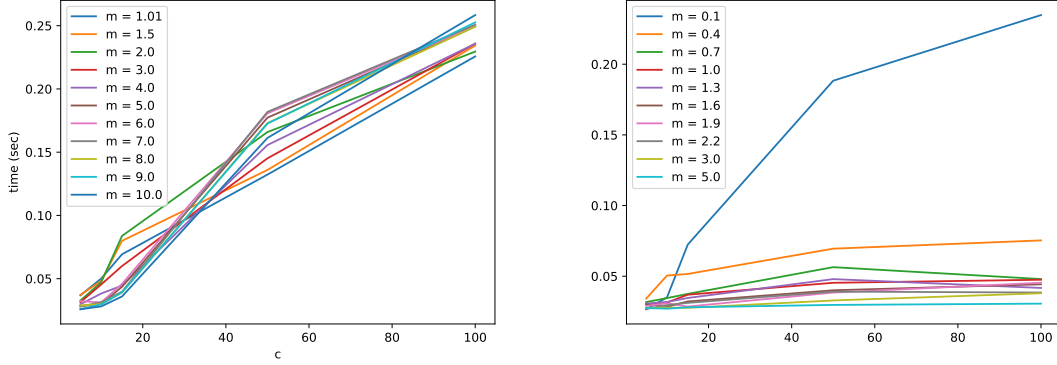


Fig. 14. Tables of average running times checking the existence of one solution for $\tau_{loc}^{Max-Min}$ (left) and $\tau_{loc}^{Max-Mean}$ (right) (in second). Total run of $R = 100$.

4.5 Values and thresholds: upper bounds, lower bounds, and solutions

From our experimental work, the statistics provide useful information about the relation between local constraints and global solutions.

Application-wise, an important question is the adjustment of lower and upper values for the thresholds τ_c , in order to get useful information. In the sequel, we explain how to infer such values via the choice of the multiplicative factor m , for the case of $\tau_{loc}^{Max-mean}$ – that of $\tau_{loc}^{Max-min}$ is similar.

The upper bound is generally application dependent. Large values of τ_c are such that most colorings are valid, yet, such large τ_c may be irrelevant application-wise. For example, transposing our Delaunay-Voronoi setting to molecular modeling, large intersection volumes between atomic representations (the equivalent of the intersection area used in this work) directly translate into prohibitive van der Waals interaction energies [23] (which explains interactions of subunits from physical aspects).

To choose a lower bound, one seeks the smallest value of m yielding the desired value of $s_{c,m}$. In our experiments, for $s_{c,m} = 1$, $m \leq 0.7$ suffices whatever the value of c (the right plot of Figure 12). For $s_{c,m} = s_0 (< 1)$, relevant pairs (c, m) are obtained by seeking the smallest values of m and c such that $s_{c,m} \geq s_0$.

In addition, we provide Figure 13 which shows the density of conflict graphs corresponding to pairs (c, m) . In fact, the statistics of density is opposite to the one of $s_{c,m}$ (i.e. for a fixed c , when m increases, a conflict graph is clearly sparser and it is the contrary for a fixed m and when c increases). Regarding pairs (c, m) whose $s_{c,m} = 1$, the statistics show that the density is at most 47% when $(c = 15, m = 4)$ for $\tau_{loc}^{Max-min}$ and at most 78% when $(c = 100, m = 0.1)$ for

$\tau_{loc}^{Max-mean}$. And the density expects to increase when c increases (but m does not) for $\tau_{loc}^{Max-min}$ but it keeps stable for $\tau_{loc}^{Max-mean}$ when c increases. In this latter case, a high fraction of conflict edges does not jeopardize the existence of solutions, as evidenced e.g. by the pair $(c = 100, m = 0.1)$ (fraction of conflict edges ~ 0.8 and $s_{100,0.1} = 1$).

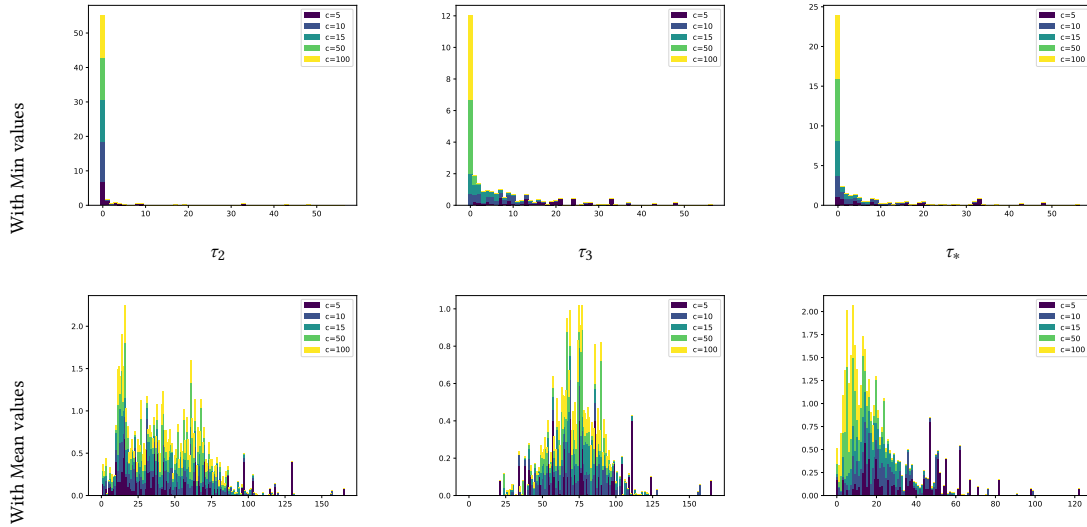


Fig. 15. **Distribution of intersection areas $ia(\cdot, \cdot)$ for Min (top row) and Mean values (bottom row).** The three plots correspond to τ_2 , τ_3 and τ_* , respectively. A total of $R = 100$ random instances were used. Note that y -axis presents the average number of edges, faces or stars of an instance; and x -axis infers values $\lfloor ia(\cdot, \cdot) \rfloor$.

5 CONCLUSION AND PERSPECTIVES

Computer Science. We have shown that the coloring problems are very hard in most graph families $\mathcal{B}(k, m)$. For a family \mathcal{B} such that \mathcal{B} -FULFILL- k -COLORING is NP-complete and \mathcal{B} -CONFLICT- k -COLORING is P, we can exploit the latter to build an algorithm for solving the former. Assume that \mathcal{B} has the following property: *each color of one vertex typically pairs up with only one color of the other*, this yields the conflict graph which is a complete bipartite graph minus a matching. Let \mathcal{A} the set of graphs obtained from $K_{k,k}$ by deleting a matching. One can show that \mathcal{A} -CONFLICT COLORING is polynomial-time solvable. Hence, although \mathcal{A} -FULFILL k -COLORING is NP-complete, when the number q of fulfilled edges is fixed, the problem can be solved in polynomial time. Indeed, for each of the $O(m^q)$ possible subgraphs F of G , one can test in polynomial time if F admits a k -coloring with no conflict edges. A natural question is then whether this problem is *Fixed-Parameter Tractable* (FPT) with respect to q , that is if it can be solved in time $f(q)n^c$ with f a computable function and c a constant. Similarly, for any fixed number q , one can decide in polynomial time whether a graph G has a coloring fulfilling all edges but at most q by testing each of the $O(m^q)$ possible subgraphs F obtained from G by removing at most q edges. It would be interesting to determine if it is FPT with respect to q .

Computational Structural biology. A central motivation for this work is the reconstruction of atomic models of macro molecular assemblies, based on (i) the interaction graph between subunits, and (ii) one set of conformations (colors) for each subunit. Two subunits in contact must exhibit a high steric complementarity, with in particular little overlap

between atoms – that would otherwise result in prohibitive (van der Waals) interaction energies. For this reason, the resulting conflict k -graphs are almost complete bipartite conflict k -graphs. Interestingly, we have noticed that in processing instances defined from the Delaunay-Voronoi setting, which mimics subunits with controlled overlap, a high fraction of conflict edges does not jeopardize the existence of solutions. Our work is therefore compelling in this setting, to process cases with of the order of tens of subunits, each exhibiting up to hundreds of conformations. In the longer run, the ability to enumerate all solutions may serve as a proxy for the free energy of the assembly, which depends on the population statistics of all valid conformations.

REFERENCES

- [1] CGAL, Computational Geometry Algorithms Library. (????). <http://www.cgal.org>.
- [2] D. Agarwal, J. Araujo, C. Caillouet, F. Cazals, D. Coudert, and S. Pérennes. 2013. Connectivity Inference in Mass Spectrometry based Structure Determination. In *European Symposium on Algorithms (Springer LNCS 8125)*, H.L. Bodlaender and G.F. Italiano (Eds.). Springer, Sophia Antipolis, France, 289–300. <http://hal.inria.fr/hal-00849873>
- [3] D. Agarwal, C. Caillouet, D. Coudert, and F. Cazals. 2015. Unveiling Contacts within Macro-molecular assemblies by solving Minimum Weight Connectivity Inference Problems. *Molecular and Cellular Proteomics* 14 (2015), 2274–2282. <https://doi.org/10.1074/mcp.M114.047779>
- [4] Anton Bernshteyn. 2019. The Johansson-Molloy theorem for DP-coloring. *Random Structures and Algorithms* 54, 4 (2019), 653–664. <https://doi.org/10.1002/rsa.20811> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/rsa.20811>
- [5] J.-D. Boissonnat and M. Yvinec. 1998. *Algorithmic geometry*. Cambridge University Press, UK. Translated by H. Brönnimann.
- [6] C.-C. Cheng and S. F. Smith. 1997. Applying constraint satisfaction techniques to job shop scheduling. *Annals of Operations Research* (1997), 327 – 357.
- [7] Zdenek Dvořák and Luke Postle. 2017. Correspondence coloring and its application to list-coloring planar graphs without cycles of lengths 4 to 8. *Journal of Combinatorial Theory, Series B* 129 (09 2017).
- [8] A. Fersht. 1999. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. Freeman.
- [9] Pierre Fraigniaud, Marc Heinrich, and Adrian Kosowski. 2016. Local Conflict Coloring. In *Proceeding of FOCS'2016*. 625–634.
- [10] Michael R. Garey, David S. Johnson, and Larry Stockmeyer. 1976. Some simplified NP-complete graph problems. *Theoretical Computer Science* 1, 3 (1976), 237 – 267. [https://doi.org/10.1016/0304-3975\(76\)90059-1](https://doi.org/10.1016/0304-3975(76)90059-1)
- [11] F.O. Hadlock. 1975. Finding a Maximum Cut of a Planar Graph in Polynomial Time. *SIAM J. Comput.* 4 (09 1975), 221–225. <https://doi.org/10.1137/0204019>
- [12] Alexander K Hartmann and Martin Weigt. 2006. *Phase transitions in combinatorial optimization problems: basics, algorithms and statistical mechanics*. John Wiley & Sons.
- [13] Narendra Jussien, Guillaume Rochart, and Xavier Lorca. 2008. Choco: an open source java constraint programming library. In *Proceeding of OSSICP'08*. 1–10.
- [14] Richard M. Karp. 1972. *Reducibility among Combinatorial Problems*. Springer US.
- [15] Seog-Jin Kim and Kenta Ozeki. 2019. A note on a Brooks’ type theorem for DP-coloring. *Journal of Graph Theory* 91, 2 (2019), 148–161. <https://doi.org/10.1002/jgt.22425> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/jgt.22425>
- [16] K. Lasker, M. Topf, A. Sali, and H. Wolfson. 2009. Inferential optimization for simultaneous fitting of multiple components into a cryoEM map of their assembly. *Journal of molecular biology* 388, 1 (2009), 180–194.
- [17] R.H. Lathrop. 1994. The protein threading problem with sequence amino acid interaction preferences is NP- complete. *Protein Engineering* 7 (1994), 1059–1068.
- [18] Runrun Liu, Sarah Loeb, Yuxue Yin, and Gexin Yu. 2019. DP-3-coloring of some planar graphs. *Discrete Mathematics* 342, 1 (2019), 178 – 189. <https://doi.org/10.1016/j.disc.2018.09.025>
- [19] T. Lozano-Perez and L. Kaelbling. 2014. A constraint-based method for solving sequential manipulation planning problems. *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2014), 3684–3691.
- [20] CPLEX User’s Manual. 1987. Ibm ilog cplex optimization studio. Version 12 (1987), 1987–2018.
- [21] V.-H. Nguyen. 2021. *PhD thesis: Graph problems motivated by (low and high) resolution models of large protein assemblies*.
- [22] Rob Phillips, Jane Kondev, Julie Theriot, Hernan G Garcia, and Nigel Orme. 2012. *Physical biology of the cell*. Garland Science.
- [23] A.J. Stone. 2002. *The Theory of Intermolecular Forces*. Oxford University Press.
- [24] J. Xu, F. Jiao, and B. Berger. 2005. A tree-decomposition approach to protein structure prediction. In *Proc IEEE Comput Syst Bioinform Conf*.
- [25] J. Xu, M. Li, D. Kim, and Y. Xu. 2003. RAPTOR: optimal protein threading by linear programming. *Journal of Bioinformatics and Computational Biology* 1 (2003), 95–117.